

REFOCUS: Training-Free Attention Sparsification via Uniform-Reference Filtering

Fengyuan Fan

Tencent

smarsufan@tencent.com

Rena Gao*

University of Melbourne

rena.gao@unimelb.edu.au

Yanbei Jiang

Meituan

jiangyanbei@meituan.com

Abstract

Dense softmax attention assigns non-zero mass to every visible key, allowing irrelevant context tokens to dilute value aggregation. We introduce **REFOCUS**, a training-free inference-time sparsifier that removes post-softmax entries below a row-local uniform-reference threshold and renormalizes the surviving mass. **REFOCUS** requires no retraining, prompt changes, or cross-step statistics. Across three dense 4–8B backbones and six metrics, it improves 15 of 18 model-benchmark cells, ties two, and incurs one -0.09 point regression. It also transfers to Qwen3-30B-A3B, improving HumanEval and WinoGrande by 0.61 and 0.47 points. Under five-seed stochastic decoding on Qwen3-4B, **REFOCUS** wins 23 of 25 seed-benchmark comparisons, and bootstrap confidence intervals for all five benchmarks exclude zero. Results on HELMET at 8k further show two improvements, two ties, and one small regression across five application-oriented categories. These findings support uniform-reference filtering as a simple accuracy-oriented modification to dense attention; the current unfused implementation does not claim inference speedups.

1 Introduction

Transformer-based language models rely on multi-head attention (Vaswani et al., 2017), where every query token attends to all keys in its causal context via a softmax operator. By construction, softmax assigns strictly positive probability mass to every visible key, regardless of relevance to the current prediction. In realistic prompts, spanning chain-of-thought reasoning, code synthesis, and long-document tasks, only a small fraction of keys carry genuine signal (Xiao et al., 2024; Sun et al., 2024); the rest consist of filler tokens, formatting characters, repetitions, and stale intermediate context.

Because the softmax is dense by design, probability mass that would otherwise concentrate on informative keys spreads across this low-signal tail, diluting the attention output and, we argue, degrading downstream accuracy (Liu et al., 2024a).

Prior work on sparse attention (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020) addresses this by restricting each query to a structured subset of keys. These methods are motivated primarily by *efficiency*: reducing the $\mathcal{O}(n^2)$ cost of full attention for long sequences. As a result, they typically require retraining or architectural changes, and accuracy preservation is treated as a constraint rather than a goal. A complementary line of work prunes the KV cache at inference time (Zhang et al., 2023; Li et al., 2024; Liu et al., 2023), but these methods track attention statistics across decoding steps and operate at the KV-eviction level rather than modifying the per-step softmax distribution.

We ask a different question: can post-softmax sparsification, without any training, *improve* downstream accuracy by removing the diffuse tail of task-irrelevant attention mass? We introduce **REFOCUS**, a training-free, plug-and-play attention sparsifier that applies a single per-row threshold to the post-softmax distribution and renormalizes the surviving entries. The threshold is length-normalized: for each row, it is set as a global pruning strength divided by the number of causally visible keys in that row. This keeps the rule comparable across short and long contexts while using only one global hyperparameter. The threshold has a clean uniform-reference reading: it removes any key whose post-softmax mass falls below a fixed fraction of the per-key mass under the row-local maximum-entropy (uniform) baseline, i.e., keys that carry less mass than this uniform baseline allots them. A mild constraint on the global scalar structurally guarantees that the row argmax always survives, keeping the renormalization denomina-

*Equal contribution.

tor strictly positive without any numerical floor.

We evaluate REFOCUS across heterogeneous short- and long-context tasks to test whether this simple rule generalizes beyond a single benchmark family. Empirical results show a consistent non-regression trend relative to dense attention and improvements on tasks where evidence is dispersed across many weakly relevant tokens. We also observe stronger gains on long-context aggregation settings, supporting the core hypothesis that filtering sub-uniform attention mass reduces noise in value integration. Importantly, REFOCUS remains lightweight in practice: it can be enabled in HuggingFace through a custom `attn_implementation="refocus"` forward, without retraining. Contributions of this work include:

- **Method.** We propose **REFOCUS**, a training-free and plug-and-play post-softmax sparsification rule that applies a length-normalized, row-wise threshold and renormalizes surviving entries. The method uses a single global hyperparameter and requires no retraining or architecture modification.
- **Principle and guarantees.** We give a uniform-reference interpretation of the threshold and show why renormalization remains well-defined under a mild constraint on the threshold scale. We further analyze key design choices (per-row vs. global thresholding and geometric vs. data-driven visible-key counting) and quantify their empirical impact (§3, §6).
- **Empirical evidence.** We provide a cross-task and cross-model evaluation covering short-context reasoning/coding tasks and long-context retrieval/aggregation benchmarks, complemented by five-seed sampling, training-free baselines, MoE transfer, and application-oriented long-context evaluation.

2 Related Work

2.1 Sparse Attention and Inference-Time Pruning

A large body of work reduces the quadratic cost of full attention by restricting each query to a structured subset of keys. Pattern-based methods combine local windows with global tokens (Beltagy et al., 2020; Zaheer et al., 2020) or use strided and fixed sparsity patterns (Child et al., 2019). Re-

former (Kitaev et al., 2020) approximates full attention via locality-sensitive hashing. These methods require pretraining or fine-tuning on the target sparse pattern and address efficiency rather than per-step attention quality. Linear attention and Performer instead approximate the softmax operator with kernelized feature maps (Katharopoulos et al., 2020; Choromanski et al., 2021); unlike REFOCUS, they replace the dense attention computation itself.

The closest training-free alternatives modify attention at inference time. Top- θ Attention (Berestizshevsky et al., 2025) calibrates per-head thresholds to approximate a fixed top- k retention target and introduces compensation terms to limit output drift. Attention calibration (ACT) suppresses hidden attention sinks on heads selected using task-specific calibration data (Yu et al., 2024). REFOCUS differs in using one calibration-free, length-normalized threshold on every row. We compare these methods directly in Section 5.4.

A related phenomenon is the *attention sink*: a small number of initial tokens consistently accumulate disproportionate attention mass across layers and heads (Xiao et al., 2024; Yu et al., 2024; Sun et al., 2024). StreamingLLM (Xiao et al., 2024) exploits this to enable infinite-context decoding by retaining sink tokens and a local window. H₂O (Zhang et al., 2023) and ScissorHands (Liu et al., 2023) identify important tokens via cumulative attention scores and evict the rest from the KV cache. SnapKV (Li et al., 2024), PyramidKV (Cai et al., 2024), and FastGen (Ge et al., 2024) refine KV eviction with observation-window pooling, layer-wise budget allocation, and head-adaptive policies, respectively. Yehezkel et al. (2024) show that sparsification mechanisms can be exploited adversarially, underscoring the sensitivity of attention distributions to targeted perturbations. These methods operate on the KV cache across decoding steps and require cross-step statistics, and are orthogonal to KV quantization approaches such as KIVI (Liu et al., 2024b); REFOCUS instead filters the per-step post-softmax distribution with no memory of prior steps and no learned components.

2.2 Long-Context Evaluation

Standard short-context benchmarks are insufficient to characterize sparse attention behavior, since the low-signal tail of attention mass grows with context length. The Needle-in-a-

Haystack (NIAH) family (Kamradt, 2023) probes single-item retrieval from long distractor contexts but does not stress evidence aggregation. RULER (Hsieh et al., 2024) provides a more comprehensive synthetic suite with controlled distractor density, covering retrieval, aggregation, variable tracking, and QA under flexible context lengths—tasks that stress the attention distribution in qualitatively different ways. HELMET (Yen et al., 2025) complements synthetic diagnostics with seven application-oriented categories, including retrieval-augmented generation, reranking, citation, and in-context learning. We use RULER at 4k and 8k for controlled analysis and HELMET at 8k to test transfer to more realistic long-context applications.

3 REFOCUS: Uniform-Reference Attention Sparsification

A central practical challenge for sparse attention is preserving model quality while removing low-value attention mass. REFOCUS is designed for this setting as Figure 1 shows: it is a post-softmax, inference-time rule that can be inserted into existing dense attention code without retraining. Intuitively, the method treats each attention row as a probability budget, removes entries that are too weak relative to a row-local baseline, and redistributes that budget over the remaining entries.

3.1 Threshold Design and Sparsification Steps

Let $S \in \mathbb{R}^{B \times H \times L_q \times L_k}$ denote scaled and causally masked logits, and let $A = \text{softmax}(S)$ denote dense attention. REFOCUS applies a row-wise operator to A : it computes a row-dependent threshold, suppresses entries below that threshold, and renormalizes the remaining mass. This keeps the downstream interface unchanged while reducing the diffuse low-probability tail.

Step 1: per-row uniform-reference threshold.

For each query row i , let L_i^{eff} be the number of causally visible keys. Under a pure causal mask with cached prefix length $L_k - L_q$, we have

$$L_i^{\text{eff}} = (L_k - L_q) + i + 1, \quad i \in \{0, \dots, L_q - 1\}. \quad (1)$$

The threshold is then defined as

$$\tau_i = \frac{\alpha}{L_i^{\text{eff}}}, \quad \alpha \in [0, 1). \quad (2)$$

We use $\alpha = 0.5$ in main experiments. Row-wise normalization is important: early rows (short visible context) are not over-pruned, while later rows (longer visible context) use proportionally smaller absolute cutoffs.

Step 2: drop sub-uniform keys. Define the keep mask

$$m_{b,h,i,j} = \mathbb{I}[A_{b,h,i,j} \geq \tau_i], \quad (3)$$

broadcast across b, h for each row threshold τ_i . The filtered distribution is

$$\hat{A}_{b,h,i,j} = A_{b,h,i,j} \cdot m_{b,h,i,j}. \quad (4)$$

This removes the diffuse tail while preserving higher-confidence entries.

Step 3: row-wise renormalization. The final REFOCUS distribution is the row-normalized version of $\hat{A}$:

$$\tilde{A}_{b,h,i,j} = \frac{\hat{A}_{b,h,i,j}}{\sum_{j'} \hat{A}_{b,h,i,j'}}. \quad (5)$$

Attention output is then computed as usual, $O = \tilde{A}V$. Therefore, REFOCUS is a drop-in replacement at the post-softmax stage rather than a new attention operator. Algorithm 1 summarizes the full per-layer forward pass.

The largest entry in any row is at least the row average. Under causal masking, that average is $1/L_i^{\text{eff}}$. Since REFOCUS uses $\tau_i = \alpha/L_i^{\text{eff}}$ with $\alpha < 1$, the row maximum must be strictly above threshold, so at least one key always survives. Therefore, the denominator in Eq. (5) is strictly positive and no numerical fallback is needed. Figure 2 illustrates this on a real attention row: the bulk of entries lie well below τ_i but carry negligible probability mass, so renormalization remains numerically stable while a small set of dominant keys absorbs the redistributed budget.

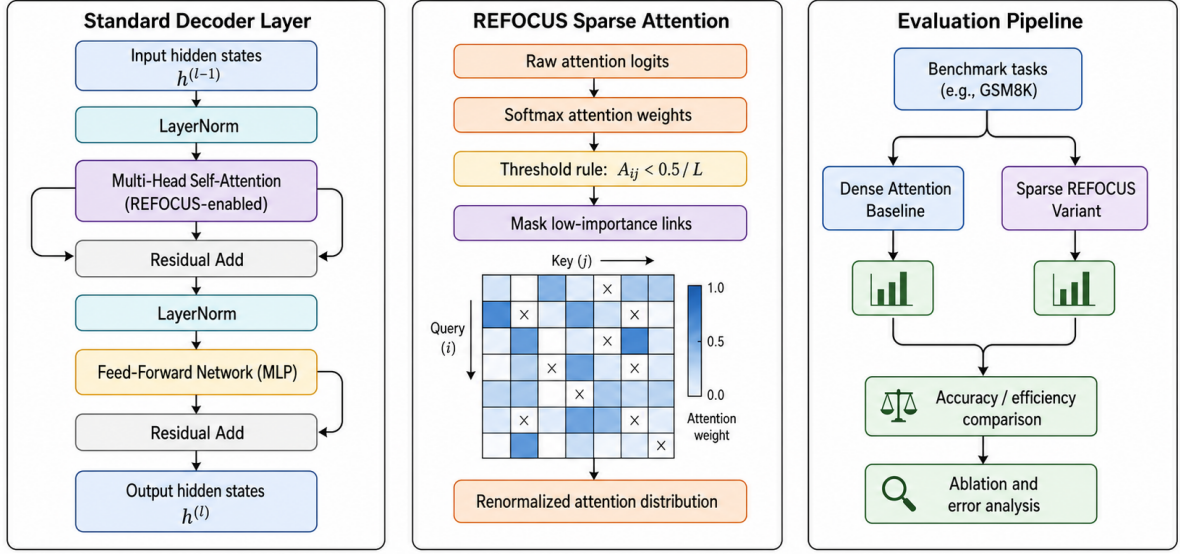
Equivalence to a re-softmax on biased logits.

Step 2 + Step 3 are mathematically equivalent (up to floating-point precision) to subtracting a large constant C on the pre-softmax logits at the dropped positions and re-softmaxing:

$$\tilde{A} = \text{softmax}(S - C \cdot (1 - m)). \quad (6)$$

We adopt the post-softmax form (Eq. 5) because it is one extra elementwise comparison, multiplication and row-sum on top of the dense path—no second softmax call, no additional logit-tensor materialization—and because it makes the per-row τ_i trivially broadcastable.

REFOCUS: Attention Pruning inside Transformer Decoding



Low-probability attention links are suppressed before final normalization.

Figure 1: Overview of the proposed REFOCUS attention pruning framework. The method is integrated into the self-attention module of a standard Transformer decoder layer, where low-probability attention links are identified by a thresholding rule, masked before normalization, and compared against a dense-attention baseline through downstream task evaluation and ablation analysis.

Algorithm 1 REFOCUS attention forward (per layer, per head)

Input: Scaled pre-softmax logits $\mathbf{S} \in \mathbb{R}^{B \times H \times L_q \times L_k}$, causal mask, value matrix $\mathbf{V}$, scalar $\alpha \in [0, 1]$

Output: Attention output $\mathbf{O}$

- 1: $\mathbf{A} \leftarrow \text{softmax}(\mathbf{S} + \text{causal_mask})$ $\triangleright$ dense softmax; masked entries = 0
- 2: **if** $\alpha > 0$ **then**
- 3: $L_i^{\text{eff}} \leftarrow (L_k - L_q) + i + 1, \quad i = 0, \dots, L_q - 1$ $\triangleright$ Eq. (1)
- 4: $\tau_i \leftarrow \alpha / L_i^{\text{eff}}$ $\triangleright$ per-row threshold, Eq. (2)
- 5: $\hat{\mathbf{A}}_{b,h,i,j} \leftarrow \mathbf{A}_{b,h,i,j} \cdot \mathbf{1}[\mathbf{A}_{b,h,i,j} \geq \tau_i]$ $\triangleright$ zero sub-uniform keys
- 6: $\tilde{\mathbf{A}}_{b,h,i,j} \leftarrow \hat{\mathbf{A}}_{b,h,i,j} / \sum_{j'} \hat{\mathbf{A}}_{b,h,i,j'}$ $\triangleright$ row renorm., Eq. (5)
- 7: **else**
- 8: $\tilde{\mathbf{A}} \leftarrow \mathbf{A}$ $\triangleright \alpha = 0$ recovers dense eager
- 9: **end if**
- 10: **return** $\mathbf{O} = \tilde{\mathbf{A}} \mathbf{V}$

3.2 Design Choices and Guarantees

Per-row, not global, length normalization. A global threshold $\tau = \alpha / L_k$ shared across all rows is misaligned with the uniform-reference reading under causal masking: row i 's uniform reference is $1/L_i^{\text{eff}}$, not $1/L_k$, so for early rows α/L_k falls well below the natural per-key scale and almost no entries get pruned. The per-row form (Eq. 2) is the principled choice and adds exactly one arange and one division to the forward pass.

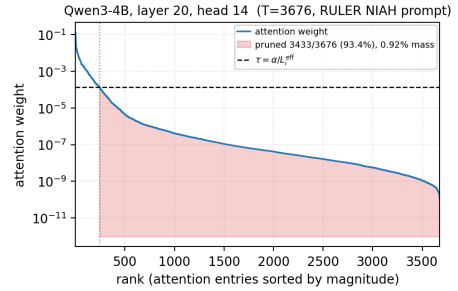


Figure 2: Post-softmax attention weights from a single head (layer 20, head 14) of Qwen3-4B on a 3,676-token RULER NIAH prompt, sorted by magnitude (log y -axis). The black dashed line is the REFOCUS threshold $\tau_i = \alpha / L_i^{\text{eff}} \approx 1.36 \times 10^{-4}$ (with $\alpha = 0.5$); entries below τ_i (red shading, to the right of the red dotted rank cutoff) are zeroed by REFOCUS and their mass is redistributed to the survivors via row renormalization. **93.4% of entries fall below τ_i , yet account for only 0.92% of the total probability mass**—the long-tail structure that makes noise pruning near-lossless.

Geometric vs. data-driven count of L_i^{eff} . A natural alternative to Eq. 1 is to count visible keys from tensor values, $L_i^{\text{eff}} = \#\{j : A_{b,h,i,j} > 0\}$. Under pure causal masking (no padding), the two are mathematically equivalent. In mixed precision (fp16/bf16), however, very small but valid probabilities can underflow to exact zero, which causes

the data-driven count to underestimate visibility, inflate τ_i , and over-prune entries. Empirically, on HumanEval with Qwen3-4B at $\alpha = 0.5$, the geometric form scores 78.05 vs. 76.83 for the data-driven form. We therefore use Eq. 1 as the default in this paper. For masking schemes beyond pure causal masking (e.g., padding or sliding windows), a data-driven visible-set definition should be used.

Length-normalized, not absolute. Using $\tau_i = \alpha/L_i^{\text{eff}}$ instead of an absolute constant ensures the pruning aggressiveness self-calibrates as the context grows. At $L_i^{\text{eff}} = 100$ we keep entries above 0.005; at $L_i^{\text{eff}} = 8000$ we keep entries above 6.25×10^{-5} . Because the average per-key mass under any attention distribution over L_i^{eff} visible keys is by definition $1/L_i^{\text{eff}}$, an absolute threshold (e.g., 0.01) would be far too aggressive at long contexts and far too lax at short contexts.

No additional learnable parameters. REFOCUS introduces exactly one hyperparameter α and zero trainable weights. It can therefore be inserted into any pretrained transformer at inference time with no additional training or fine-tuning. $\alpha = 0$ recovers dense eager attention exactly (the keep mask becomes all-ones), so the same checkpoint can be served in both modes by toggling a single scalar.

3.3 Implementation

REFOCUS is implemented as a single self-contained attention forward (refocus_attention.py) and registered into HuggingFace Transformers’ attention dispatch table. Once registered, loading a model with `attn_implementation="refocus"` routes attention through the REFOCUS forward without requiring any model-code edits. The runtime sparsification strength is controlled through a single global scalar α .

Inside the forward pass, the sparsification logic is intentionally minimal: after the dense causal softmax, the method computes a per-row visible-key count, derives $\tau_i = \alpha/L_i^{\text{eff}}$, masks probabilities below the threshold, and renormalizes each row. This preserves the dense pipeline structure while adding only lightweight elementwise operations and one row-wise sum/division. A reference implementation of the core sparsification block is given in Appendix B.

Three implementation details are worth flagging. (i) The per-row τ_i tensor has shape

$[1, 1, L_q, 1]$ and is broadcast across B, H at no memory cost. (ii) The closed-form arange captures both the prefill regime ($L_q = L_k$, i.e., $\tau_i = \alpha/(i + 1)$) and the incremental decoding regime ($L_q = 1$, i.e., a single $\tau = \alpha/L_k$) without a code branch. (iii) HuggingFace materializes a 4-D additive causal mask only for the named "eager" forward; custom forwards including REFOCUS receive `attention_mask=None` and must construct the additive causal bias themselves. The reference implementation includes a `short_build_causal_mask` helper for $L_q > 1$ and skips it on $L_q = 1$ (which is trivially causal).

4 Experimental Setup

Table 1 summarizes the models and benchmarks used in this work.

Category	Item	Description
Model	Qwen3-4B (Team, 2025)	Primary model in nothink mode (<code>enable_thinking=False</code>); also used for multi-seed robustness tests.
Model	Dense 8B backbones	Qwen3-8B and Llama-3.1-8B-Instruct, evaluated under the same greedy protocol.
Model	Qwen3-30B-A3B (Team, 2025)	Mixture-of-experts transfer setting with 30B total and 3B active parameters per token.
Benchmark	GSM8K (Cobbe et al., 2021)	Grade-school arithmetic reasoning (chain-of-thought style evaluation).
Benchmark	HumanEval (Chen et al., 2021)	Python function synthesis benchmark (<code>pass@1</code>).
Benchmark	WinoGrande (Sakaguchi et al., 2020)	Commonsense pronoun resolution (binary choice).
Benchmark	MMLU-STEM (Hendrycks et al., 2021)	Multiple-choice multitask QA, STEM subset (19 subjects, 3,153 items).
Benchmark	RULER (Hsieh et al., 2024) (<code>ctx ∈ {4096, 8192}</code>)	Long-context suite with 13 sub-tasks across NIAH retrieval, aggregation (<code>fwe+cwe</code>), variable tracking (VT), and QA families.
Benchmark	HELMET (Yen et al., 2025) (<code>ctx = 8192</code>)	Five application-oriented categories with reference-based metrics; LongQA and summarization are omitted because they require an external model judge.

Table 1: Models and benchmarks used in the experimental setup. All benchmarks are evaluated on full test splits without sub-sampling.

Baseline and REFOCUS configuration.

The dense baseline uses HuggingFace’s standard `attn_implementation="eager"` forward. REFOCUS is enabled via `attn_implementation="refocus"` with $\alpha = 0.5$ for all main-table results. Both configurations share identical model weights, prompts formatted with the chat template, and decoding hyperparameters (`do_sample=False`); the only difference is the attention forward. Because decoding is deterministic, no random seed is required and each greedy result is a single full-split run.

Stochastic robustness protocol. To quantify sampling variability, we additionally evaluate Qwen3-4B with temperature=0.6 and top_p=0.95 using five paired seeds (42, 123, 2024, 7, and 314). Dense and REFOCUS use the same seed for each comparison. We report means and standard deviations across seeds, paired one-sided tests, and paired bootstrap 95% confidence intervals for the accuracy difference. Alternative sampling settings are reported in Appendix C.

5 Results

We compare REFOCUS with the dense eager baseline under identical prompts and decoding settings. Deltas are score-point differences on full test splits or the category-specific metrics indicated below.

5.1 Short-Context Benchmarks

REFOCUS improves 15 of 18 model–benchmark cells, ties two, and has one -0.09 pp regression. Qwen3-4B improves on all tasks ($+1.21$ pp on average), with its largest gains on HumanEval ($+1.83$) and WinoGrande ($+1.81$); Qwen3-8B also improves throughout ($+0.80$ pp on average). Llama-3.1-8B gains on three tasks, ties two, and slightly declines on RULER-4k.

The compact-model result is strongest: Qwen3-4B with REFOCUS matches dense Qwen3-8B on HumanEval without changing the checkpoint or decoding budget. The same α also transfers to RULER, averaging $+0.46$ and $+0.37$ pp at 4k and 8k. Because these are deterministic full-split evaluations, we treat the differences as descriptive effect sizes. Further interpretation of model scale, denoising, and long-context behavior appears in Appendix A.

5.2 Robustness under Stochastic Decoding

REFOCUS wins 23 of 25 paired seed–benchmark comparisons, with one tie and one loss; all paired bootstrap intervals exclude zero. Gains are largest on RULER-4k and WinoGrande, while GSM8K remains positive across all seeds. Five seeds limit test resolution; further statistics appear in Appendix C.

5.3 Transfer to a Mixture-of-Experts Model

REFOCUS applies unchanged to Qwen3-30B-A3B and improves HumanEval and WinoGrande by $+0.61$ and $+0.47$ pp, respectively. This sup-

ports initial MoE transfer, but only across two tasks.

5.4 Comparison with Training-Free Alternatives

Tuned attention top- p achieves the best Qwen3-4B WinoGrande score, but becomes worse than Dense at $p = 0.7$; temperature sharpening is similarly sensitive. All tested REFOCUS thresholds improve over Dense. On Llama-3.1-8B, REFOCUS improves without calibration, while Top- θ degrades and ACT is a no-op. Thus, REFOCUS is a stable calibration-free rule, not always the best tuned single-task transformation.

5.5 Long-Context Evaluation: RULER at 4k and 8k

RULER gains are concentrated in aggregation: REFOCUS improves this family by $+7.10$ pp at 4k and $+3.70$ pp at 8k, driving overall gains of $+1.32$ and $+0.64$ pp. NIAH and variable tracking remain near ceiling, while QA changes by only $+0.10$ and -0.10 pp. The result therefore supports improved long-context aggregation, not a universal gain across long-context behaviors.

5.6 Application-Oriented Long Context: HELMET

HELMET shows two improvements, two ties, and one small regression. The largest gain is $+0.52$ on many-shot ICL, while citation drops 0.20 . These results support preservation rather than uniform improvement. LongQA and summarization are omitted because they require an external model judge.

6 Analysis

6.1 Effect of α

We sweep $\alpha \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$ on Qwen3-4B to characterize the sensitivity of REFOCUS to its single hyper-parameter. Table 8 reports test-set accuracy on GSM8K, HumanEval, and RULER-4k together with the model-wide pruning ratio induced by each α (the layer-wise breakdown is deferred to Fig. 3). Three observations are worth highlighting.

Strict non-regression across the entire sweep.

Every value of α from 0.10 up to 0.90 matches or exceeds the dense baseline on all three benchmarks. This is a much stronger property than

Model	Config	Short-Context Accuracy (%)				Long-Context (%)	
		GSM8K	HumanEval	WinoGrande	MMLU-STEM	RULER-4k	RULER-8k
Qwen3-4B	Dense (eager)	86.28	76.22	58.88	74.12	86.00	82.91
	REFOCUS ($\alpha=0.5$)	86.88	78.05	60.69	75.17	87.32	83.55
Qwen3-8B	Dense (eager)	88.78	78.05	64.01	77.48	92.74	88.51
	REFOCUS ($\alpha=0.5$)	89.46	80.49	64.64	78.05	92.89	88.85
Llama-3.1-8B-Instruct	Dense (eager)	78.70	60.98	55.88	67.81	95.35	92.02
	REFOCUS ($\alpha=0.5$)	78.70	60.98	57.30	67.84	95.26	92.15

Table 2: Accuracy on full test splits. Dense and REFOCUS share identical weights, prompts, tokenizers and greedy decoding settings; the only difference is attn_implementation. Bold indicates the higher value within each dense/REFOCUS pair, with ties left unbolded. HumanEval reports pass@1; RULER reports strict full-match across all 13 sub-tasks at $\text{ctx} \in \{4096, 8192\}$; all other columns report exact-match or multiple-choice accuracy.

Benchmark	Dense (%)	REFOCUS (%)	Δ (pp)	Bootstrap 95% CI	W/T/L
RULER-4k	84.27 $\pm$ 0.19	86.19 $\pm$ 0.24	+1.91	[+1.62, +2.21]	5/0/0
WinoGrande	58.66 $\pm$ 0.22	60.54 $\pm$ 0.09	+1.88	[+1.70, +2.07]	5/0/0
HumanEval	76.10 $\pm$ 1.17	77.20 $\pm$ 0.70	+1.10	[+0.37, +2.07]	4/1/0
MMLU-STEM	73.97 $\pm$ 0.34	74.84 $\pm$ 0.55	+0.87	[+0.27, +1.42]	4/0/1
GSM8K	86.87 $\pm$ 0.48	87.26 $\pm$ 0.24	+0.39	[+0.12, +0.74]	5/0/0

Table 3: Five-seed results on Qwen3-4B with temperature sampling ($T = 0.6$, $p = 0.95$). Values are mean $\pm$ standard deviation over paired seeds. Confidence intervals are paired bootstrap intervals for the REFOCUS minus Dense difference. W/T/L counts seed-level wins, ties, and losses.

Configuration	HumanEval	WinoGrande
Qwen3-30B-A3B, Dense	80.49	69.22
Qwen3-30B-A3B, REFOCUS	81.10	69.69
Δ (pp)	+0.61	+0.47

Table 4: Greedy-decoding transfer results on the Qwen3-30B-A3B mixture-of-experts model.

“ $\alpha = 0.5$ happens to work”: the per-row threshold $\tau_i = \alpha/L_i^{\text{eff}}$ behaves as a robust knob rather than a fragile hyper-parameter.

Aggressive pruning is compatible with high accuracy. At $\alpha = 0.5$ the model already zeroes out 81.9% of post-softmax entries on average—and the best GSM8K accuracy (87.79, $\alpha \in \{0.75, 0.90\}$) occurs at pruning ratios above 85%, where less than one in seven entries remain non-zero. This empirically supports the central premise of the paper: the dense softmax distributes mass over many sub-uniform entries that carry no useful signal, and removing them is information-preserving rather than information-destroying. As Fig. 3 will show, this behaviour is not localised to a few outlier layers: the per-layer pruning ratio stays above 42% in every single layer even at $\alpha = 0.10$, so the sub-uniform tail is a global property of the attention distribution.

Decoding becomes shorter, not longer. The average number of generated output tokens on GSM8K monotonically decreases from 245.9 (dense) to 231.1 ($\alpha = 0.90$)—a reduction of about 6%—while accuracy increases. REFOCUS therefore does not pay for its accuracy gain with longer chain-of-thought; on the contrary, removing sub-uniform tail mass appears to make the model commit to its answer slightly faster.

The fact that GSM8K and RULER-4k keep improving through aggressive settings while HumanEval peaks at $\alpha = 0.5$ confirms that the task-wise optimum is not universal. We use $\alpha = 0.5$ as a conservative global default because it maximizes HumanEval while improving both other tasks and all six Qwen3-4B metrics in Table 2. If validation data are available, the one-dimensional five-point sweep in Table 8 provides a low-cost tuning procedure; otherwise, $\alpha = 0.5$ avoids the short-context regression risk observed at larger values while retaining long-context gains.

6.2 Per-Layer Pruning Ratio

Figure 3 decomposes the model-wide **Prune %** column of Table 8 into a per-layer profile. Two patterns are visible. First, for every α the curve is monotonically non-decreasing in depth on aggregate: early layers retain a denser attention pat-

Backbone	Method	Setting	WinoGrande (%)	Δ (pp)	Calibration
Qwen3-4B	Dense attention	–	58.88	–	No
	REFOCUS	$\alpha = 0.1$	59.43	+0.55	No
	REFOCUS	$\alpha = 0.5$	60.69	+1.81	No
	REFOCUS	$\alpha = 0.9$	59.98	+1.10	No
	Attention top- p	$p = 0.9$	61.40	+2.52	No
	Attention top- p	$p = 0.7$	57.62	−1.26	No
	Temperature sharpening	$T = 0.85$	60.22	+1.34	No
	Temperature sharpening	$T = 0.5$	49.57	−9.31	No
Llama-3.1-8B	Dense attention	–	55.88	–	No
	REFOCUS	$\alpha = 0.5$	57.30	+1.42	No
	Top- θ (post-softmax VMC)	$k = 64$	55.49	−0.39	Yes
	ACT	$\beta = 0.4$	55.88	0.00	Yes

Table 5: Greedy WinoGrande comparison with training-free attention transformations. Attention top- p truncates and renormalizes the attention distribution rather than the output-token distribution. Top- θ and ACT use their calibration procedures; ACT selected no beneficial heads in this Llama setting. The full top- p /temperature sweep is in Appendix D.

Configuration	RULER ctx=4096 (full-match %)					RULER ctx=8192 (full-match %)				
	NIAH	Agg.	VT	QA	Avg	NIAH	Agg.	VT	QA	Avg
Qwen3-4B, Dense (eager)	97.02	50.90	100.0	70.00	86.00	97.23	34.10	99.80	66.00	82.91
Qwen3-4B w/ REFOCUS ($\alpha = 0.5$)	97.38	58.00	100.0	70.10	87.32	97.38	37.80	99.80	65.90	83.55
Δ (pp)	+0.36	+7.10	0.00	+0.10	+1.32	+0.15	+3.70	0.00	−0.10	+0.64

Table 6: RULER results on Qwen3-4B at $\text{ctx} \in \{4096, 8192\}$ (greedy decoding, full split, 6,500 examples per context). Columns aggregate the 13 sub-tasks into the four standard RULER families: **NIAH** (8 needle-in-a-haystack variants), **Agg.** (cwe, fwe aggregation), **VT** (variable tracking) and **QA** (qa_1, qa_2). All numbers are strict full-match accuracy; **Avg** is the micro-average across all 6,500 examples. REFOCUS matches or exceeds Dense on every family except a single −0.10 pp on QA at $\text{ctx} = 8192$.

Category	Metric	Dense	REFOCUS	Δ
Recall	substr. EM	100.00	100.00	0.00
RAG	substr. EM	65.00	65.00	0.00
Rerank	NDCG@10	69.65	69.95	+0.30
Cite	6-metric avg.	31.22	31.02	−0.20
ICL	exact match	74.12	74.64	+0.52

Table 7: HELMET results on Qwen3-4B at 8k context with greedy decoding and $\alpha = 0.5$. RAG averages four datasets, citation averages six metrics, and ICL averages five datasets. Metrics are heterogeneous and are therefore not averaged across categories.

tern, while deep layers concentrate on a smaller set of task-relevant positions and therefore expose a larger sub-uniform tail to REFOCUS—the $\alpha = 0.5$ curve, for instance, rises from roughly 69% in the first layer to $\sim 90\%$ in the deepest layers. This is consistent with the qualitative observation that early-layer attention patterns are typically broader (Michel et al., 2019) while later layers become increasingly selective. Second, the spacing between successive α curves shrinks rapidly: increasing α from 0.5 to 0.9 adds only ~ 5 pp of

α	Accuracy (%)			Prune %
	GSM8K	HumanEval	RULER-4k	
0 (dense)	86.28	76.22	86.00	0.00
0.10	86.50	76.83	86.14	61.84
0.25	86.50	76.83	86.35	74.05
0.50	86.88	78.05	87.32	81.91
0.75	87.79	76.22	87.89	85.81
0.90	87.79	77.44	88.06	87.35

Table 8: α ablation on Qwen3-4B. **Prune %** is the model-wide average fraction of post-softmax entries zeroed per row, measured on 64 chat-templated GSM8K prefill prompts; the per-layer breakdown is plotted in Fig. 3. Bold marks the best accuracy in each column; $\alpha = 0$ recovers exact dense eager attention.

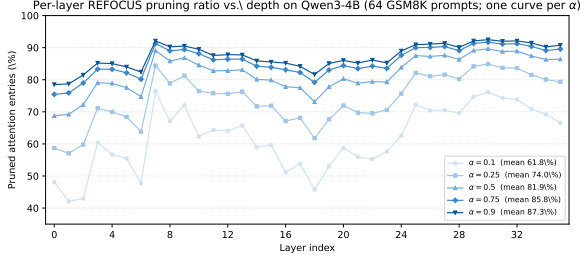


Figure 3: Per-layer pruning ratio on Qwen3-4B as a function of α , averaged over 64 chat-templated GSM8K prefill prompts. Each curve corresponds to one $\alpha \in \{0.10, 0.25, 0.50, 0.75, 0.90\}$; the x -axis is the transformer layer index (early $\rightarrow$ deep) and the y -axis is the fraction of post-softmax entries zeroed per row, averaged across heads, query positions and prompts.

extra pruning on average, indicating that the sub-uniform tail mass that REFOCUS targets is concentrated tightly below uniform and is largely already removed at $\alpha = 0.5$. This layer-wise variation also motivates future work on per-layer α schedules, where deep layers are pushed harder than early layers without a single global α having to compromise between the two regimes.

7 Conclusion

We introduced REFOCUS, a training-free post-softmax sparsifier that applies the length-normalized threshold $\tau_i = \alpha / L_i^{\text{eff}}$ and renormalizes each attention row. The rule removes sub-uniform attention mass, while $\alpha < 1$ guarantees that the row argmax survives. It requires only a minimal, HuggingFace-compatible modification to dense attention.

Across three dense backbones, REFOCUS improves 15 of 18 model–benchmark cells, ties two, and has one -0.09 pp regression; it also improves both evaluated Qwen3-30B-A3B tasks. Five-seed results are positive across all five Qwen3-4B benchmarks, while HELMET indicates preservation rather than universal gains. Although tuned attention top- p can achieve a higher single-task optimum, REFOCUS is more stable and requires no calibration. These results support it as an accuracy-oriented modification, not a speedup method. Future work includes longer contexts, broader MoE and multilingual evaluation, fused sparse kernels, and per-layer α schedules.

Limitations

We evaluate three dense 4–8B backbones and one 30B-total/3B-active MoE model, but the MoE study covers only HumanEval and WinoGrande, and detailed long-context analysis is limited to Qwen3-4B. We do not test multilingual tasks or contexts beyond 8k, and HELMET excludes judge-dependent LongQA and summarization. The results therefore support the filtering principle rather than fully characterize its behavior across scales and applications.

The main results use greedy decoding; five-seed sampling is limited to Qwen3-4B, and training-free baselines are compared mainly on WinoGrande, so conclusions may change across tasks or decoding strategies. The unfused implementation retains the dense attention matrix, provides no memory reduction, and adds $1.50\text{--}1.92\times$ prefill latency, approaching $1.04\times$ dense latency only for 8k decode; speedups require a fused sparse kernel. Finally, row-wise thresholding may remove individually weak tokens that are collectively important. Although a single global α is robust in our ablations, task- or layer-specific thresholds may be necessary in such settings.

References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- Konstantin Berestizshevsky, Renzo Andri, and Lukas Cavigelli. 2025. Top-theta attention: Sparsifying transformers by compensated thresholding. *arXiv preprint arXiv:2502.08363*.
- Zefan Cai, Yichi Zhang, Biqing Qi, Yuliang Liu, Beiqing Hou, Bowen Zhou, and Wen Xiao. 2024. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Rahul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.
- Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Łukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Łukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekish, Fei Jia, and Boris Ginsburg. 2024. [RULER: What’s the real context window size of your long-context language models?](#) In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Greg Kamradt. 2023. Needle in a haystack – pressure testing llms. *Technical report*.
- Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*.
- Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.
- Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianhao Feng, Patrick Lewis, and Deming Chen. 2024. SnapKV: LLM knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024a. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhao Xu, Anastasios Kyrilidis, and Anshumali Shrivastava. 2023. ScissorHands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test time. *arXiv preprint arXiv:2305.17118*.
- Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhao Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024b. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8732–8740.
- Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. In *Conference on Language Modeling (COLM)*.
- Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *International Conference on Learning Representations*.

Oryan Yehezkel, Alon Zolfi, Amit Baras, Yuval Elovici, and Asaf Shabtai. 2024. [DeSparsify: Adversarial attack against token sparsification mechanisms](#). In *Advances in Neural Information Processing Systems*.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to evaluate long-context language models effectively and thoroughly. In *International Conference on Learning Representations*.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. 2024. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. In *Proceedings of the 41st International Conference on Machine Learning*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H₂O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*.

A Additional Discussion of Model Scale and Long-Context Behavior

Compact-model contribution. Improving a small backbone usually requires additional training, a larger decoding budget, or a larger model. REFOCUS instead keeps the checkpoint, prompts, and greedy decoding fixed. Qwen3-4B improves across arithmetic, code, commonsense, multiple-choice, and both RULER context lengths; on HumanEval, it reaches the dense Qwen3-8B score of 78.05 pass@1. This does not imply that sparsification replaces scaling, but it suggests that part of the compact model’s performance gap comes from mixing irrelevant context into the value stream rather than from capacity alone.

Denoising interpretation. For larger or saturated models, removing low-probability mass may primarily preserve accuracy. For smaller models, the same operation can act as a lightweight denoising step: limited capacity may make value aggregation more sensitive to diffuse attention tails. The stronger Qwen3-4B gains are consistent with this interpretation, although they do not establish a causal mechanism.

Long-context behavior. The same global α improves average RULER accuracy without retuning, changing the KV cache, or modifying prompts. The family-level gains are highly concentrated: NIAH and variable tracking are already near ceiling, QA is nearly unchanged, and aggregation improves by +7.10 pp at 4k and +3.70 pp at 8k. Aggregation requires combining evidence across many distractors, where flatter attention leaves more sub-uniform mass available to filter. We therefore interpret REFOCUS as improving long-context aggregation in this setting, not as uniformly improving every long-context behavior.

B Reference Implementation

For completeness we include the minimal post-softmax filter used in all experiments. Given the dense causal attention probabilities, REFOCUS removes entries below the per-row threshold $\tau_i = \alpha / L_i^{\text{eff}}$ and renormalizes the surviving mass. The block below is verbatim from our HuggingFace-compatible attention backend; setting $\alpha = 0$ skips the branch entirely and recovers dense eager attention exactly, while $\alpha < 1$ keeps $\tau_i < 1 / L_i^{\text{eff}}$, guaranteeing that the row argmax always survives so the renormalization denominator stays positive.

```
# attn_weights = softmax(QK^T / sqrt(d) +
# causal_mask), shape [B, H, Lq, Lk].
# Row i is causally visible to (k_len - q_len)
# + i + 1 keys.
if alpha > 0.0:
    valid_count = torch.arange(
        k_len - q_len + 1, k_len + 1,
        device=attn_weights.device, dtype=
            attn_weights.dtype,
    ).view(1, 1, q_len, 1)
    tau = alpha / valid_count
    attn_weights = attn_weights.masked_fill(
        attn_weights < tau, 0.0)
    attn_weights = attn_weights / attn_weights.
        sum(dim=-1, keepdim=True)
```

C Stochastic-Decoding Details

Table 9 reports the paired tests underlying Table 3. Tests are one-sided in the hypothesized direction. The paired t -test does not cross 0.05 on HumanEval, and the Wilcoxon test does not cross 0.05 on MMLU-STEM; we therefore do not claim that every task is significant under every test. The paired bootstrap intervals are positive for all five tasks.

The RULER-4k gain also persists across the four sampling configurations in Table 10, showing that it is not specific to one temperature/nucleus pair.

D Alternative-Baseline Sweep

For attention top- p , we sort each attention row, retain the smallest prefix whose cumulative mass reaches p , and renormalize. Temperature sharpening rescales the attention logits before softmax. Table 11 shows that both alternatives have useful task-specific settings but are more sensitive to aggressive hyperparameters than REFOCUS.

E Local Smoothing Variant

To test whether span-level evidence can protect individually weak tokens, we replace each probability used in the keep decision by a symmetric moving average while retaining the original probability for renormalization. A window of three gives a small additional WinoGrande gain, but this variant has been evaluated on only one task and is not part of the default method.

F Full RULER-4k Breakdown

G Prototype Efficiency

We benchmark attention forwards for the Qwen3-4B configuration (batch size 2, 32 query heads,

Benchmark	Δ (pp)	Paired t p	Wilcoxon p	Bootstrap 95% CI
RULER-4k	+1.91	0.000042	0.031	[+1.62, +2.21]
WinoGrande	+1.88	0.000031	0.031	[+1.70, +2.07]
HumanEval	+1.10	0.052	0.033	[+0.37, +2.07]
MMLU-STEM	+0.87	0.019	0.063	[+0.27, +1.42]
GSM8K	+0.39	0.043	0.031	[+0.12, +0.74]

Table 9: One-sided paired tests across five stochastic-decoding seeds.

Sampling configuration	Dense	REFOCUS	Δ
$T = 0.6, p = 0.95$	84.27	86.19	+1.91
$T = 0.6, p = 0.90$	84.54	86.28	+1.74
$T = 1.0, p = 0.90$	83.29	84.74	+1.45
$T = 1.0, p = 0.95$	83.06	84.31	+1.25

Table 10: Qwen3-4B RULER-4k mean accuracy across five seeds under alternative sampling configurations.

Method	WinoGrande	Δ
Dense attention	58.88	–
REFOCUS ($\alpha = 0.1$)	59.43	+0.55
REFOCUS ($\alpha = 0.3$)	59.51	+0.63
REFOCUS ($\alpha = 0.5$)	60.69	+1.81
REFOCUS ($\alpha = 0.7$)	59.91	+1.03
REFOCUS ($\alpha = 0.9$)	59.98	+1.10
Attention top- p ($p = 0.95$)	59.12	+0.24
Attention top- p ($p = 0.90$)	61.40	+2.52
Attention top- p ($p = 0.80$)	59.35	+0.47
Attention top- p ($p = 0.70$)	57.62	–1.26
Attention top- p ($p = 0.50$)	41.20	–17.68
Temperature ($T = 0.85$)	60.22	+1.34
Temperature ($T = 0.70$)	59.19	+0.32
Temperature ($T = 0.50$)	49.57	–9.31
Temperature ($T = 0.25$)	29.76	–29.12

Table 11: Full Qwen3-4B WinoGrande sweep for calibration-free attention transformations under greedy decoding.

Window	Accuracy	Prune ratio	Δ
Dense	58.88	–	–
1 (default)	60.69	90.3%	+1.81
3	61.01	87.0%	+2.13
5	60.69	84.7%	+1.81
7	59.43	83.0%	+0.55

Table 12: Local-smoothing ablation on Qwen3-4B WinoGrande.

8 KV heads, head dimension 128, float16) on an NVIDIA A100. Each number averages 20 timed runs after 5 warmup iterations. These measurements characterize the unfused PyTorch implementation; they are not end-to-end model throughput measurements.

REFOCUS has the same measured peak GPU memory as Dense in this benchmark ($1.00\times$ at every length) because it modifies the materialized attention matrix in place. It is slower during prefill and approaches parity only for long-KV decode. A fused sparse attention–value kernel may exploit the induced zeros, but that optimization and any resulting speedup remain future work.

Sub-task	Qwen3-4B			Qwen3-8B		
	Dense	REFOCUS	Δ	Dense	REFOCUS	Δ
fwe	68.80	76.00	+7.20	87.80	88.60	+0.80
cwe	33.00	40.00	+7.00	91.00	92.20	+1.20
niah_single_3	92.80	96.00	+3.20	100.00	100.00	0.00
niah_multivalue	85.40	86.00	+0.60	87.80	87.00	-0.80
niah_multikey_1	99.60	100.00	+0.40	100.00	100.00	0.00
niah_single_2	99.40	99.60	+0.20	100.00	100.00	0.00
qa_2	61.20	61.40	+0.20	60.40	59.80	-0.60
niah_multikey_2	100.00	100.00	0.00	100.00	100.00	0.00
niah_multiquery	99.40	99.40	0.00	99.20	99.40	+0.20
qa_1	78.80	78.80	0.00	79.40	80.60	+1.20
vt	100.00	100.00	0.00	100.00	100.00	0.00
niah_single_1	100.00	99.60	-0.40	100.00	100.00	0.00
niah_multikey_3	99.60	98.40	-1.20	100.00	100.00	0.00

Table 13: Complete greedy RULER-4k sub-task results. Qwen3-4B improves on 7 sub-tasks, ties on 4, and declines on 2; Qwen3-8B improves on 4, ties on 7, and declines on 2.

Prefill length	Dense	REFOCUS	Top- p	Temp.	Top- k	RF/Dense	Top- p /Dense	Temp./Dense	Top- k /Dense
512	0.39	0.67	1.73	0.42	1.31	$1.72\times$	$4.45\times$	$1.06\times$	$3.36\times$
1024	1.14	2.18	4.98	1.24	3.82	$1.92\times$	$4.37\times$	$1.09\times$	$3.36\times$
2048	8.16	12.22	23.16	8.53	15.77	$1.50\times$	$2.84\times$	$1.05\times$	$1.93\times$
4096	23.53	39.53	84.90	25.03	48.44	$1.68\times$	$3.61\times$	$1.06\times$	$2.06\times$
8192	77.48	145.68	454.91	83.49	178.72	$1.88\times$	$5.87\times$	$1.08\times$	$2.31\times$

Table 14: Prefill attention latency in milliseconds. Top- p uses $p = 0.9$, temperature uses $T = 0.85$, and top- k uses the corresponding implementation from the baseline benchmark.

KV length	Dense	REFOCUS	Top- p	Temp.	Top- k	RF/Dense	Top- p /Dense	Temp./Dense	Top- k /Dense
512	0.10	0.17	0.20	0.11	0.15	$1.69\times$	$1.95\times$	$1.08\times$	$1.43\times$
1024	0.11	0.28	0.20	0.11	0.15	$2.54\times$	$1.83\times$	$1.01\times$	$1.35\times$
2048	0.19	0.22	0.26	0.19	0.23	$1.15\times$	$1.37\times$	$1.01\times$	$1.22\times$
4096	0.35	0.39	0.44	0.35	0.41	$1.09\times$	$1.25\times$	$1.01\times$	$1.15\times$
8192	0.68	0.71	0.84	0.68	0.75	$1.04\times$	$1.24\times$	$1.01\times$	$1.11\times$

Table 15: Single-token decode attention latency in milliseconds as the KV-cache length varies.